PostScript Operators

Operator: add

num1 num2 **add** *num3*

This operator returns the addition of the two arguments.

- stackunderflow
- typecheck
- undefinedresult

See also:

- div
- mul
- sub

Operator: arc

x-coord y-coord r ang1 ang2 **arc** -

This operator adds an arc to the current path. The arc is generated by sweeping a line segment of length *r*, and tied at the point (*x-coord y-coord*), in a counter-clockwise direction from an angle *ang1* to an angle *ang2*. Note: a straight line segment will connect the current point to the first point of the arc, if they are not the same.

- limitcheck
- stackunderflow
- typecheck

Operator: begin

dict **begin** -

This operator pushes the dictionary *dict* onto the dictionary stack. Where it can be used for def and name lookup. This operator allows an operator to set up a dictionary for its own use (e.g. for local variables).

Errors:

- dictstackoverflow

- invalidaccess
- stackunderflow
- typecheck

Operator: bind

procedure1 **bind** *procedure2*

The bind operator goes through *procedure1* and replaces any operator names with their associate operators. Names which do not refer to operators are left alone. Operators within *procedure1* which have unrestricted access will have bind called on themselves before they are inserted into the procedure. The new procedure with operators instead of operator names is returned on the stack as *procedure2*.

The main effect and use of this operator is to reduce the amount of name lookup done by the interpreter. This speeds up execution and ties down the behavior of operators.

Errors:

- typecheck

Operator: clip

- clip -

This operator intersects the current clipping path with the current path and sets the current clipping path to the results. Any part of a path drawn after calling this operator which extends outside this new clipping area will simply not be drawn. If the given path is open, clip will treat it as if it were closed. Also, clip does not destroy the current path when it is finished... it may be used for other activities.

It is important to note that there is no easy way to restore the clip path to a larger size once it has been set. The best way to set the clip path is to wrap it in a gsave and grestore pair.

Errors:

- limitcheck

Operator: closepath

- closepath -

This operator adds a line segment to the current path from the current point to the first point in the path. This closes the path so that it may be filled.

Errors:

- limitcheck

Also see the following operators:

- **newpath**
- **moveto**
- **lineto**

Operator: charpath

string bool **charpath -**

This operator takes the given string and appends the path which the characters define to the current path. The result is can be used as any other path for stroking, filling, or clipping.

The boolean argument informs charpath what to do if the font was not designed to be stoked. If the boolean is true, the path will be modified to be filled and clipped (but not stroked). If the boolean is false, the path will be suitable to be stroked (but not filled or clipped).

- limitcheck
- nocurrentpoint
- stackunderflow
- typecheck

See also:

- clip
- fill
- show
- stroke

Operator: curveto

x1 y1 x2 y2 x3 y3 **curveto -**

This operator draws a curve from the current point to the point (*x3, y3*) using points (*x1, y1*) and (*x2, y2*) as control points. The curve is a Bézier cubic curve. In such a curve, the tangent of the curve at the current point will be a line segment running from the current point to (*x1, y1*) and the tangent at (*x3, y3*) is the line running from (*x3, y3*) to (*x2, y2*).

- limitcheck
- nocurrentpoint
- stackunderflow

- typecheck

See also:

- arc
- lineto
- moveto

Operator: def

name value **def -**

This operator associates the *name* with *value* in the dictionary at the top of the dictionary stack. This operator essentially defines names to have values in the dictionary and is used to define variables and operators.

Errors:

- dictfull
- invalidaccess
- limitcheck
- stackunderflow
- typecheck
- VMerror

Operator: div

num1 num2 **div** *num3*

This operator returns the result of dividing *num1* by *num2*. The result is always a real.

- stackunderflow
- typecheck
- undefinedresult

See also:

- add
- mul
- sub

Operator: dup

object **dup** *object object*

This operate pushes a second copy of the topmost object on the operand stack. If the object is a reference to an array, string, or similar composite object, only the reference is duplicated; both references will still refer to the same object.

See also:

- exch
- index
- pop

Errors:

- stackoverflow
- stackunderflow

Operator: end

- end -

This operator pops the topmost dictionary off of the dictionary stack. The dictionary below it becomes the new current dictionary.

Errors:

- dictstackunderflow

Operator: exch

value1 value2 **exch** *value2 value1*

This operator simply exchanges the top two items on the operand stack. It does not matter what the operands are.

See also:

- dup
- index
- pop

Errors:

- stackunderflow

Operator: fill

- fill -

This operator closes and fills the current path with the current color. Any ink within the path is obliterated. Note that fill blanks out the current path as if it had called newpath. If you want the current path preserved, you should use gsave and grestore to preserve the path.

Errors:

- limitcheck

Operator: findfont

name **findfont** *font*

This operator looks for the named font in the font dictionary. If it finds the font, it pushes the font on the stack for later processing. It signals an error if the font can not be found.

Errors:

- invalidfont
- stackunderflow
- typecheck

Also see the following operators:

- **scalefont**
- **setfont**

Operator: for

initial increment limit proc **for -**

This operator will execute *proc* repeatedly. The first time *proc* is executed, it will be given *initial* as the top operand. Each time it is executed after that, the top operand will be incremented by *increment*. This process will continue until the argument would have exceeded *limit*.

- stackoverflow
- stackunderflow
- typecheck

See also:

- if
- ifelse

Operator: grestore

- grestore -

Sets the current graphics state to the topmost graphics state on graphics state stack and pops that state off the stack. This operator is almost always used in conjunction with gsave.

Operator: gsave

- gsave -

This operator pushes a copy of the current graphics state onto the graphics state stack. The graphics state consists of (among other things):

- Current Transformation Matrix
- Current Path
- Clip Path
- Current Color
- Current Font
- Current Gray Value

gsave is typically used with grestore whenever you need to change the graphics state temporarily and return to the original.

Errors:

- limitcheck

Operator: if

bool proc **if -**

This operator will execute *proc* if *bool* is true.

- stackunderflow
- typecheck

Operator: ifelse

bool proc1 proc2 **ifelse** -

This operator will execute *proc1* if *bool* is true and *proc2* otherwise.

- stackunderflow
- typecheck

Operator: index

value_n ... value_0 n **index** *value_n ... value_0 value_n*

This operator grabs the nth item off the operand stack (item 0 is the one just under the index you push on the stack for the operator) and pushes it on top of the stack.

See also:

- dup
- exch
- pop

Errors:

- rangecheck
- stackunderflow
- typecheck

Operator: lineto

x-coord y-coord **lineto** -

This operator adds a line into the path. The line is from the current point to the point (*x-coord y-coord*). After the line is added to the path, the current point is set to (*x-coord y-coord*). It is an error to call **lineto** without having a current point.

Errors:

- limitcheck
- nocurrentpoint
- stackunderflow
- typecheck

Also see the following operators:

- **rlineto**
- **moveto**
- **rmoveto**
- **curveto**
- **arc**
- **closepath**

Operator: moveto

x-coord y-coord **moveto** -

This operator moves the current point of the current path to the given point in user space. If a **moveto** operator immediately follows another **moveto** operator, the previous one is erased.

Errors:

- limitcheck
- stackunderflow
- typecheck

Also see the following operators:

- **rmoveto**
- **lineto**
- **curveto**
- **arc**
- **closepath**

Operator: mul

value1 value2 **mul** *product*

This operator multiplies the first two operands on the stack and pushes the result back onto the stack. The result is an integer if both operands are integers and the product is not out of range. If the product is too big, or one of the operands is a real, the result will be a real.

Errors:

- stackunderflow
- typecheck
- undefinedresult

Operator: newpath

- newpath -

The newpath operator clears out the current path and prepares the system to start a new current path. This operator should be called before starting *any* new path, even though some operators call it implicitly.

Operator: pop

value **pop -**

This operator just removes the top-most item off of the operand stack.

See also:

- dup
- exch
- index

Errors:

- stackunderflow

Operator: restore

state **restore -**

This restores the total state of the PostScript system to the state saved in *state*.

Errors:

- invalidrestore
- stackunderflow
- typecheck

See also:

- save

Operator: rlineto

dx dy **rlineto -**

This operator adds a line into the path. The line is from the current point to a point found by adding *dx* to the current *x* and *dy* to the current *y*. After line is added to the path, the current point is set to the new point. It is an error to call **lineto** without having a current point.

Errors:

- limitcheck
- nocurrentpoint
- stackunderflow
- typecheck

Also see the following operators:

- **lineto**
- **moveto**
- **rmoveto**
- **curveto**
- **arc**
- **closepath**

Operator: rmoveto

dx dy **rmoveto** -

This operator moves the current point of the current path by adding *dx* to the current *x* and *dy* to the current *y*.

Errors:

- limitcheck
- stackunderflow
- typecheck

Also see the following operators:

- **moveto**
- **lineto**
- **curveto**
- **arc**
- **closepath**

Operator: rotate

angle **rotate** -

This operator has the effect of rotating the user space counter-clockwise by *angle* degrees (negative angles rotate clockwise). The rotation occurs around the current origin.

- rangecheck
- stackunderflow
- typecheck

See also:

- scale
- translate

Operator: save

- save *state*

This operator gathers up the complete state of the PostScript system and saves it in *state*. Errors:

- limitcheck
- stackoverflow

See Also:

- restore

Operator: scale

sx sy **scale -**

This operator has the effect of scaling the user coordinates. All coordinates will be multiplied by *sx* in the horizontal direction, and *sy* in the vertical.

The origin will not be affected by this operation.

- rangecheck
- stackunderflow
- typecheck

See also:

- rotate
- translate

Operator: scalefont

font size **scalefont** *font*

This operator takes the given font and scales it by the given scale factor. The resulting scaled font is pushed onto the stack. A *size* of one produces the same sized characters as the original font, 0.5 produces half-size characters, and so on.

Errors:

- invalidfont
- stackunderflow
- typecheck
- undefined

Also see the following operators:

- **findfont**
- **setfont**

Operator: setfont

font **setfont** -

This operator sets the current font to be *font*. This font can be the result of any font creation or modification operator. This font is used in all subsequent character operations like show.

- invalidfont
- stackunderflow
- typecheck

Also see:

- findfont
- scalefont

Operator: setgray

gray-value **setgray** - This operator sets the current intensity of the ink to *gray-value*. *gray-value* must be a number from 0 (black) to 1 (white). This will affect all markings stroked or filled onto the page. This applies even to path components created before the call to **setgray** as long as they have not yet been stroked.

- stackunderflow

- typecheck
- undefined

Operator: setlinewidth

width **setlinewidth** - This operator sets the width of all lines to be stroked to *width*, which must be specified in points. A line width of zero is possible and is interpreted to be a hairline, as thin as can be rendered on the given device.

- stackunderflow
- typecheck

Operator: show

string **show -**

This operator draws the given string onto the page. The current graphics state applies, so the current font, fontsize, gray value, and current transformation matrix all apply.

The location for the text is set by the current point. The current point will specify the leftmost point of the baseline for the text.

- invalidaccess
- invalidfont
- nocurrentpoint
- rangecheck
- stackunderflow
- typecheck

See also:

- charpath
- moveto
- setfont

Operator: showpage

- showpage -

This operator commits the current page to print and ejects the page from printing device. **showpage** also prepares a new blank page.

Operator: stroke

- stroke -

This operator draws a line along the current path using the current settings. This includes the current line thickness, current pen color, current dash pattern, current settings for how lines should be joined, and what kind of caps they should have. These settings are the settings at the time the stroke operator is invoked.

A closed path consisting of two or more points at the same location is a degenerate path. A degenerate path will be drawn only if you have set the line caps to round caps. If your line caps are not round caps, or if the path is not closed, the path will not be drawn. If the path is drawn, it will appear as a filled circle center at the point.

Errors:

- limitcheck

Operator: sub

num1 num2 **sub** *num3*

This operator returns the result of subtracting *num2* from *num1*.

- stackunderflow
- typecheck
- undefinedresult

See also:

- add
- div
- mul

Operator: translate

x-coord y-coord **translate -**

This operator has the affect of moving the origin to the point (*x-coord*, *y-coord*) in the current user space.

- rangecheck
- stackunderflow
- typecheck

See also:

- rotate
- scale

Index of Operators

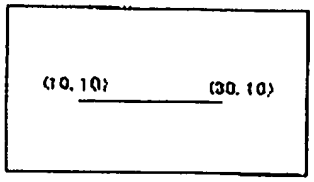
PostScriptに関する基礎事項

ここではPostScript言語を利用して基本的な描画を行うプログラムについて説明します。

○線を引くプログラム

```
newpath
10 10 moveto
30 10 lineto
1 setlinewidth
stroke
showpage
```

ソース1



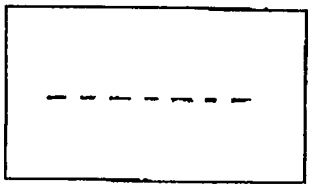
実行例1（イメージ画像）

簡単にこのプログラムの説明をしますと、まずnewpathはカレントパスを空にし、新しいパスをはじめることを宣言しています。次に(10,10)に移動し(moveto)、そこから(30,10)へ直線のパスを 作成(lineto)しています。そしてこ れから引く線の太さを設定し(setlinewidth)、strokeによってカレントパスに描画します。 最後にshowpageすることによってカレ ントページを印刷しています。

○線を引くプログラム（応用例）

```
newpath
10 10 moveto
30 10 lineto
2 setlinewidth
0.5 setgray
[4 2 2 2] 0 setdash
stroke
showpage
```

ソース2

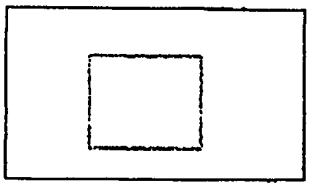


実行例2（イメージ画像）

ソース1との違いについて説明します。まず 2 setlinewidthでは、引かれる線の太さを指定しています。数字は太さを表しています。次に、0.5 setgrayでは、引かれる線の色の濃淡をあ らわしています。1が白、0が黒を表します。[4 2 2 2] 0 setdashでは引かれる線を点線に しています。[4 2 2 2]は配列を意味していますが、各数字は、長さ4 の線、長さ2の空白、長さ2の線、長さ2の空白というパターンを意味 しています。その次の0はスタートが指定したパターンのどこから始ま るかをあらわしています。

○ボックスを書く

```
newpath
10 10 moveto
30 10 lineto
30 30 lineto
10 30 lineto
closepath
0.5 setgray
stroke
```



実行例3（イメージ画像）

```
showpage
```

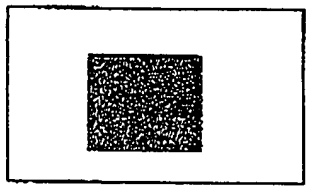
ソース3

簡単に解説します。ソースの5行目までは今までと変わ らないのでわかると思います。そこまでで四角形の1辺がたりないよう な図形になります。その次のclosepath は残りのパスを描いて、閉路を完成させる働きがあります。具体的には 書きはじめの座標と、現在の座標を結ぶようなパスを描きます。

○ボックスを書く(少し応用編)

```
newpath
10 10 moveto
30 10 lineto
30 30 lineto
10 30 lineto
closepath
0.5 setgray
fill
showpage
```

ソース4



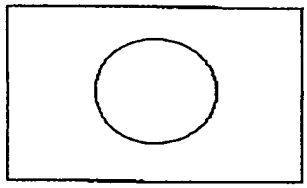
実行例4(イメージ画像)

例3と違っているのはstrokeが なくなり、fillが使われている点です。これは閉路の中を塗りつぶすオペレーターです。

○円を描く

```
newpath
20 20 10 0 360 arc
stroke
showpage
```

ソース5



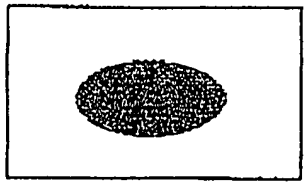
実行例5(イメージ画像)

新しくでてきた20 20 10 0 360 arcに関して解説します。このオペレーターは円の中心の座標、半径、開始位置の角度、終了の角度をもらって円弧を描くものです。同 じようなオペレーターにarcnというも のがありますが、違いは、arcは時計と逆回りに円弧を描くのに対して、arcnは時計回りに円弧を描きます。

○楕円を描く

```
newpath
2 1 scale
20 20 10 0 360 arc
fill
showpage
```

ソース6



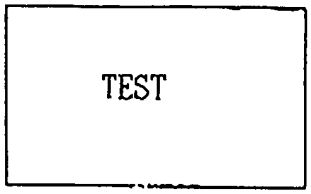
実行例6(イメージ画像)

前のプログラムと違う点は、1 2 scaleの部分です。これはX座標の値とY座標の値をそれぞれ2 倍、1倍とするというオペレーターです。これによってX座標はY座標 の2倍になるので円は横(X方向)に長い楕円が得られるわけです。

○文字を描く

```
/Times-Roman findfont 3 scalefont setfont
10 10 moveto
(TEST) show
showpage
```

ソース7



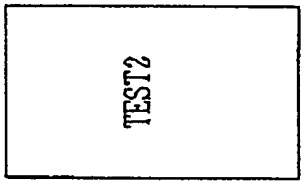
実行例7（イメージ画像）

いままでのプログラムと大きく違う点は最初にフォントの設定を 行っているところです。/Times-Roman findfont 3 scalefont setfontに関して簡単に説明すると、findfontはその前（ここでは Times_Roman）をFontDirectoryと呼ばれる辞書の中から探してきます。その後、findfontによってフォントの サイズを決定します。最後にsetfontに よってこのフォントを現在のフォントに指定しています。あとは、書き たい位置に移動し、文字列（postscriptでは( )に囲まれた文字が文字 列です。）を指定して、showにより描 いています。

○文字を描く（応用編）

```
90 rotate
/Times-Roman findfont 3 scalefont setfont
10 10 moveto
(TEST2) show
showpage
```

ソース8



実行例8（イメージ画像）

ソース7と違う点は90 rotateの 部分です。これはユーザー座標系を（この例では90度）回転させるとい うオペレーターです。ユーザー座標系とは、実際の座標系を一時的に変 更して用いる座標形のことです。ここでは回転についてあげていますが、実際は平行移動するtranslateや変換行 列を定義して回転させることもできます。

まとめと発展

ここでは簡単な図形や文字の描き方を紹介しましたが、Postscriptは他のプログラム言語と同様にデータ型や、関数のようなもの（Postscriptでは 実行型配列などといったりしますが）、ローカル変数、再帰的な表現など も使うことができます。